MICROTECHNIC
SOLUTIONS

SpryteByter™

DESIGNER SERIES

By Howard & Bill Cook

Copyright © 1984 by Microtechnic Solutions Inc.

PRELIMINARY MANUAL

You may return this manual for a final printed version. Final manuals will be available after July 1, 1984.

MICROTECHNIC SOLUTIONS, INC.
P.O. BOX 2940
NEW HAVEN, CT. 06515

COPYRIGHT NOTICE

COPYRIGHT © ***1984*** by Microtechnic Solutions, Inc. All rights reserved. No part of this publication may be reproduced in whole or in part without the prior written consent of Microtechnic Solutions, Inc.

Although every attempt has been made to verify the accuracy of this document, we cannot assume any liability for errors or omissions. No warranty or other guarantee can be given as to the suitability or accuracy of this software for a particular purpose, nor can we be liable for damage arising from the use of same. As a condition of purchase and/or use of this product, the purchaser or user accepts all risk of damage or liability arising from the use of same.

Unauthorized copying or transmitting of programs on any media is strictly prohibited. All programs are subject to copyright.

WARNING!

This product is copy-protected. Attempting to duplicate this disk may DAMAGE your disk drive!

Owners Name: _____

_____

Date Purchased:

TABLE OF CONTENTS

Introduction 1

A First Look At SpryteByter 3

Monocolor menus 6
- DRAW menu 7
- SHIFT menu 8
- ROTATE menu 9
- DATA menu 10
- REVIEW submenu 11
- COLOR menu 12

Multicolor menus 13
- DRAW menu 14
- SHIFT menu 15
- DATA menu 16
- REVIEW submenu 17

SpryteByter Notes 18
- General information 18
- The "turn" Options 19
- Sprite Storage and Sprite Files 21
- The "I/O" Options 21
- LOADing a Sprite File 21
- SAVEing a Sprite File 23
- The "data in" Options 25
- Printouts of Sprites 26
- Notes for Beginners 28
- Notes for Advanced Programmers 31

The Game Maker 32

Backups and Warranty 34

Notes 35

SpryteByter

THE SpryteByter MANUAL

INTRODUCTION

SpryteByter is perhaps the most powerful and versatile sprite development package available for any modern microcomputer. It allows users easily to create and edit sprites, both monocolor and multicolor, and makes sprite file handling a simple task. SpryteByter is ideal for the professional game developer — but the novice programmer will find it just as useful (and fun to use as well!).

The program is full-featured: almost every practical function that may be applied to sprite development has been included. SpryteByter saves hours of tedious byte-by-byte programming, and allows the user to focus his creative abilities on the sprites themselves. The Gamemaker, included in the package, provides an easy method of appending sprite and program files for maximum efficiency.

This manual is designed to demonstrate and explain the features of SpryteByter. Programming techniques are not discussed in detail — users are urged to refer to the Commodore 64 Programmer's Reference Guide for such information.

GETTING STARTED;

The SpryteByter package consists of SpryteByter and The Game Maker. Also included are two demonstration sprite files (monomovie and multimovie).

LOADing:

Make sure the disk is properly seated in the drive, and that the drive door is closed. Please note that the package is designed to use DEVICE 8 DRIVE 0 for ALL disk activities. Type LOAD"0:★",8,1 (and RETURN). SpryteByter will begin LOADing. The error light may flash erratically during the LOAD — this is normal. Disk LOADs should take about one minute to complete. The error light should remain off after the LOAD has completed.

Copyright © 1984 by Microtechnic Solutions Inc.

RUNning SpryteByter:

After LOADing, the screen will change to a neutral gray/black color format, and the DRAW menu for monocolor sprites will then appear.

Notice the general layout of the screen — this format is common throughout the various menus in SpryteByter. The bulk of the screen in occupied by a 24 X 21 grid, upon which all sprite development will take place. Immediately to the right of its design grid is the menu, which includes the options applying to the current menu as well as the available branches to other menus. The upper right-hand corner of the screen will display all four possible configurations of the sprite (normal, vertically expanded, horizontally expanded, and fully expanded); this area will remain blank until a sprite is placed into the ACCUMULATOR SPRITE, either by drawing or replacement (see "General Information", page 18).

USING SpryteByter:

Use of SpryteByter is discussed in the description of the various MENUS, which follow this section. First time users of SpryteByter should "walk through" A First Look at SpryteByter on page 3. See also SpryteByter Notes beginning on page18.

EXITING SpryteByter:

SpryteByter may only be exited by turning the computer off — the RUN/STOP and RESTORE keys will not break the program RUN. If a sprite file is in memory, SAVE it prior to shutdown — otherwise it will be lost. Disk users should make absolutely sure that no disk is present in the drive before turning the computer off.

SpryteByter was written with the assumption that the user has a good working knowledge of his computer and peripherals. The programs were written to be as "user affectionate" as possible. However, a lack of familiarity with their use could lead to possible file loss. Users are encouraged to read this manual thoroughly and develop a good working knowledge of the programs and their features prior to serious sprite development.

Copyright © 1984 by Microtechnic Solutions Inc.

A First Look at SpryteByter

When SpryteByter is run, the first menu to appear is always the DRAW menu in the monocolor mode and the box shaped cursor (which is a sprite) is in the upper lefthand corner of the design grid. If you hit the cursor keys, this cursor will move. You can also move the cursor with a joystick plugged into joystick port 2. If you hit the HOME key, the cursor will return to the upper lefthand corner of the grid.

With the cursor in the upper lefthand corner of the design grid, hit the 'A' key. The top row of plus signs in the grid will all reverse and four horizontal lines will appear in the upper righthand corner of the screen. This is the beginning of drawing a sprite. Now hit the 'B' key and a vertical line will appear on the grid and in the four sprites in the upper righthand corner of the screen.

Now move the cursor away from the corner of the grid and hit the 'C' key. A diagonal line is drawn through the cursor. Now hit the 'D' key. Move the cursor again and try the 'A', 'B', 'C' and 'D' keys. Notice that the lines are always drawn through the cursor. Any time you hit the 'E' key in this menu, the sprite will reverse. Try it.

If you hit the 'J' key, the menu will be replaced by a notice that your sprite has been stored as sprite 0. The data for your sprite has been stored in SpryteByter's memory and can be recalled later even if you change or erase the sprite. You can store as many as 184 sprites in this manner. Hit any key and the DRAW menu will reappear.

If you hit the SHIFT key and the CLR/HOME together, your sprite will dissappear and the cursor will return to the upper lefthand corner of the design grid. UNLESS YOUR SPRITE HAS BEEN STORED, IT IS COMPLETELY LOST.

Move the cursor around the grid occasionally hitting the SPACE bar. You will notice that, if you hit the space bar when the cursor is on an unreversed plus sign, the plus sign will reverse and a dot will appear on your sprites in the upper righthand corner of the screen. If you hit the space bar when the cursor is on a

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

reversed plus sign, then an unreversed plus sign will replace it and a dot will dissappear from your sprites. Now you know how to do detailed drawing from the keyboard.

If you have a joystick plugged into port 2, move the cursor with the joystick all over the grid observing what happens as you try to move off the grid. Then hit the f3 function key and repeat this experiment and notice the difference. Hitting the f3 key again will restore the original joystick movement.

Now move the cursor around using the fire button to draw with. After you have experimented for a while drawing with the joystick and fire button, hit the f1 function key and draw some more. Drawing this way is noticably different. You can revert to the original joystick/fire button drawing mode by hitting the f1 key again.

SpryteByter is much more enjoyable to work with a joystick than without one — inexpensive joysticks seem to work just as well with SpryteByter as the precision sticks.

If you hit the unSHIFTed CLR/HOME key, the cursor will return to the upper lefthand corner of the grid without erasing the spirte.

The other options using keys 'F', 'G', 'H' and 'I' in the DRAW menu branch SpryteByter to other monocolor menus which you should experiment with just as you have with the DRAW menu. Before doing so, try hitting the 'I' key which will branch SpryteByter to the monocolor COLOR menu. Notice that on this menu, there is an option

M: multiclr

If you hit 'M', SpryteByter will access the multicolor DRAW menu. Here the design grid does not have the plus signs and the background and border colors are the same. Experiment with hitting the keys 'A' through 'F' in this menu and drawing (as you did in the monocolor DRAW menu) with the checkmark in different places. The line draw options are not available in the multicolor mode because they seem more confusing than useful here. The

► ► ►

Copyright © 1984 by Microtechnic Solutions Inc.

SpryteByter

[HOME], [CLR/HOME] and f3 keys have the same effect as in the monocolor DRAW menu but there is only one joystick draw method (i.e., the f1 key has no effect). You can return to the monocolor COLOR menu by hitting the "I" key.

The SpryteByter Manual has a page detailing each of the SpryteByter menus. Experiment with them all — making sure you read appropriate sections of SpryteByter notes. In particular SpryteByter notes will explain what to do with the sprites you may have stored. DO NOT OVERLOOK THE USEFULNESS OF "THE GAME MAKER" FOR PROGRAMMING WITH SPRITES.

We hope you will enjoy SpryteByter for gamemaking and for play.

Copyright © 1984 by Microtechnic Solutions Inc.

SpryteByter Monocolor Menus

DRAW MENU

A: h. line
B: v. line
C: up diag
D: dn diag
E: reverse
F: shift
G: rotate
H: data
I: color
J: store

SHIFT MENU

A: up
B: down
C: right
D: left
E: invert
F: obvert
G: 180 turn
H: draw
I: rotate
J: store

ROTATE MENU

A: turn
B: h. sym
C: v. sym
D: pt. sym
E: all sym
F: draw
G: shift
H: data
I: color
J: store

DATA MENU

A: I/O
B: printout
C: data in
D: review
E: movie
F: draw
G: shift
H: rotate
I: color
J: store

REVIEW submenu

Sprite O
A: next
B: back 1
C: edit
D: delete
E: del abov
F: insert
G: and
H: or
I: end revu

COLOR MENU

A: sprite
B: background
C: border
D: draw
E: shift
F: rotate
G: data
H: store
M: multiclr

COLOR submenu1

A: black
B: white
C: red
D: cyan
E: purple
F: green
G: blue
H: yellow
I: other
J: no chnge

COLOR submenu2

A: orange
B: brown
C: lt red
D: gray 1
E: gray 2
F: lt green
G: lt blue
H: gray 3
I: original
J: no chnge

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

DRAW Menu for Monocolor Sprites

A: h. line
B: v. line
C: up diag
D: dn diag
E: reverse
F: shift
G: rotate
H: data
I: color
J: store

A: — draws a horizontal line through the cursor.
B: — draws a vertical line through the cursor.
C: — draws a diagonal line with positive slope through the cursor.
D: — draws a diagonal line with negative slope through the cursor.
E: — reverses the sprite — turns on pixels off, turns off pixels on.
F: — BRANCHES to SpryteByter's monocolor SHIFT menu.
G: — BRANCHES to SpryteByter's monocolor ROTATE menu.
H: — BRACHES to SpryteByter's monocolor DATA menu.
I: — BRANCHES to SpryteByter's monocolor COLOR menu.
J: — stores the sprite in SpryteByter's memory.
f1 — changes the joystick drawing operation.
f2 — stops the cursor from "wrapping around" OR reverts to "wrapping around" cursor.
[HOME] — moves the cursor to the top lefthand corner of the grid.
[CLR] — clears the sprite (turns all pixels off) and moves the cursor to the top lefthand corner of the grid.

Copyright © 1984 by Microtechnic Solutions Inc.

SHIFT Menu for Monocolor Sprites

A: up	D: left	G: 180 turn	J: store
B: down	E: invert	H: draw	
C: right	F: obvert	I: rotate	

A: — shifts the sprite up 0 to 8 pixels.
B: — shifts the spirte down 0 to 8 pixels.
C: — shifts the sprite right 0 to 8 pixels.
D: — shifts the sprite left 0 to 8 pixels.
E: — turns the sprite upside down.
F: — makes a mirror image of the sprite.
G: — turns the sprite 180 degrees.
H: — BRANCHES to SpryteByter's monocolor DRAW menu.
I: — BRANCHES to SpryteByter's monocolor ROTATE menu.
J: — stores the sprite in SpryteByter's memory.
f1 — changes the joystick drawing operation.
f2 — stops the cursor from "wrapping around" OR reverts to "wrapping around" cursor.
[HOME] — moves the cursor to the top lefthand corner of the grid.
[CLR] — clears the sprite (turns all pixels off) and moves the cursor to the top lefthand corner of the grid.

NOTE: Functions A and B prompt the user with "How many rows?", and functions C and D prompt with "How many columns?" Respond by entering a number from 0 to 8.

BRANCHING TO OTHER MENUS: The SHIFT menu is the only menu in the monocolor mode which does not allow direct branching to ALL other monocolor menus. You can branch to the monocolor DRAW and ROTATE menus in the monocolor mode by hitting keys 'H' and 'I', respectively, and then to any other menu from either of these two.

Copyright © 1984 by Microtechnic Solutions Inc.

ROTATE Menu for Monocolor Sprites

A: turn	F: draw
B: h. sym	G: shift
C: v. sym	H: data
D: pt. sym	I: color
E: all sym	J: store

A: — rotates the sprite around the cursor through any user chosen integer angle from -999 to 9999. See the note on the turn options on page19.

B: — makes the sprite symmetric with respect to the horizontal line through the cursor.

C: — makes the sprite symmetric with respect to the vertical line through the cursor.

D: — makes the sprite symmetric with respect to the cursor.

E: — accomplishes functions B, C and D.

F: — BRANCHES to Sprytebyter's monocolor draw menu.

G: — BRANCHES to SpryteByter's monocolor SHIFT menu.

H: — BRANCHES to SpryteByter's monocolor DATA menu.

I: — BRANCHES to SpryteByter's monocolor COLOR menu.

J: — stores the sprite in SpryteByter's memory.

[HOME] — moves the cursor to the center of the grid.

[CLR] — restores the sprite which was present when the ROTATE menu was entered and moves the cursor to the center of the grid.

Copyright © 1984 by Microtechnic Solutions Inc.

DATA Menu for Monocolor Sprites

A: I/O	F: draw
B: printout	G: shift
C: data in	H: rotate
D: review	I: color
E: movie	J: store

A: — allows you to save a file on tape or disk of all sprite data in SpryteByter's memory OR, if there are no sprites in SpryteByter's memory, allows you to load a sprite file into memory from tape or disk. See notes on page 21.

B: — prints a representation of the sprite and the data for the sprite if there is a printer (device #4) present. Does nothing if no printer is present. See "Printouts of Sprites" on page 26.

C: — allows you to input sprite data and draws a sprite from that data. This allows you to preview and edit sprites from magazine, or other printed, program listings. Enter data as prompted with numbers from 0 to 255 and hit the RETURN key after each entry. If the function key f4 is hit instead of the RETURN key, then the typed in number will fill the rest of the sprite data. If you have not entered a number and hit the f8 key, then all remaining data numbers will become zero. If you do not wish to enter data, press R. See notes on page 25.

D: — accesses the REVIEW submenu.

E: — makes a movie of the sprites in SpryteByter's memory. Prompts for speed — enter a number from 0 (very slow) to 9.

F: — BRANCHES to SpryteByter's monocolor DRAW menu.

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

G: — BRANCHES to SpryteByter's monocolor SHIFT menu.
H: — BRANCHES to SpryteByter's monocolor ROTATE menu.
I: — BRANCHES to SpryteByter's monocolor COLOR menu.
J: — stores the sprite in SpryteByter's memory.

REVIEW Submenu for Monocolor DATA menu

sprite 0
A: next
B: back 1
C: edit
D: delete
E: del abov
F: insert
G: and
H: or
I: end reuv

A: — advances review to next sprite in memory.
B: — goes backward one sprite in memory.
C: — moves sprite under review into the sprite accumulator and returns to the DATA menu. Allows you to alter sprites from memory.
D: — removes sprite under review from memory.
E: — removes ALL sprites above sprite under review from memory.
F: — inserts the accumulator sprite in memory in front of the sprite under review.
G: — logically ANDs the sprite under review with the accumulator sprite and places the result in the accumulator.
H: — logically ORs the sprite under review with the accumulator sprite and places the result in the accumulator.
I: — returns to the DATA menu.

Copyright © 1984 by Microtechnic Solutions Inc.

COLOR menus for Monocolor Sprites

A:	sprite	A:	black	A:	orange
B:	backgrnd	B:	white	B:	brown
C:	border	C:	red	C:	lt red
D:	draw	D:	cyan	D:	gral 1
E:	shift	E:	purple	E:	gray 2
F:	rotate	F:	green	F:	lt green
G:	data	G:	blue	G:	lt blue
H:	store	H:	yellow	H:	gray 3
M:	multiclr	I:	other	I:	original
		J:	no chnge	J:	no chnge

A: — accesses submenu for SPRITE color change.
B: — accesses submenu for BACKGROUND color change.
C: — accesses submenu for BORDER color change.
D: — BRANCHES to SpryteByter's monocolor DRAW menu.
E: — BRANCHES to SpryteByter's monocolor SHIFT menu.
F: — BRANCHES to SpryteByter's monocolor ROTATE menu.
G: — BRANCHES to SpryteByter's monocolor DATA menu.
H: — stores the sprite in SpryteByter's memory.
M: — accesses SpryteByter's MULTICOLOR mode.

NOTE: In the color menu, pressing 'A', 'B' or 'C' will access COLOR submenu1 for a color change. A choice of 'A', 'B" 'C', 'D', 'E', 'F', 'G' or 'H' will make a change to the indicated color. A choice of 'I' will access COLOR submenu2 where 'A' through 'H' causes a change to the indicated color and a choice of 'I' restores

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

SpryteByter

SpryteByter's default colors. For either submenu1 or submenu2, a choice of 'J' returns SpryteByter to the COLOR menu with no color change.

● ● ***CAUTION:*** SpryteByter's menus in the monocolor mode are the same color as the sprite. Thus, if the sprite color is made to be the same as the background color, both sprites and menu will seem to disappear. AT ANY TIME WHEN SpryteByter IS IN THE MONOCOLOR MODE, PRESSING THE SHIFT KEY AND RETURN KEY TOGETHER WILL RESTORE SpryteByter's DEFAULT COLORS.

SpryteByter Multicolor Menus

DRAW MENU

A: bgd 12
B: spr 11
C: mc0 0
D: mc1 15
E: crs 1
F: change
G: shift
H: data
I: monoclr
J: store

SHIFT MENU

A: up
B: down
C: right
D: left
E: invert
F: obvert
G: turn
H: draw
I: data
J: store

DATA MENU

A: I/O
B: printout
C: data in
D: review
E: movie
F: draw
G: shift
H: store

REVIEW submenu

Sprite 0
A: next
B: back 1
C: edit
D: delete
E: del abov
F: insert
G: end revu

Copyright © 1984 by Microtechnic Solutions Inc.

DRAW Menu for Multicolor Sprites

A: bdg 12	G: shift	D: mc1 15	J: store
B: spr 11	H: data	E: crs 1	
C: mc0 0	I: monoclr	F: change	

A: — moves the checkmark to A, allowing BACKGROUND color change.

B: — moves the checkmark to B, allowing SPRITE draw and color change.

C: — moves the checkmark to C, allowing MULTICOLOR 0 draw and color change.

D: — moves the checkmark to D, allowing MULTICOLOR 1 draw and color change.

E: — moves the checkmark to E, allowing CURSOR color change.

F: — changes color according to checkmark location.

G: — BRANCHES to SpryteByter's multicolor SHIFT menu.

H: — BRANCHES to SpryteByter's multicolor DATA menu.

I: — BRANCHES to SpryteByter's MONOCOLOR mode.

J: — stores the sprite in SpryteByter's memory.

[HOME] — moves the cursor to the top lefthand corner of the grid.

[CLR] — clears the sprite (turns all pixels off) and moves the cursor to the top lefthand corner of the grid.

NOTE: Move the cursor with the JOYSTICK or the CURSOR keys. DRAW with the JOYSTICK fire button or the SPACE BAR in the color by the checkmark. If the checkmark is by A (background) or E (cursor), only erasing of pixels can be accomplished with the joystick fire button or the SPACE bar.

BRANCHING TO OTHER MENUS: You can branch to the any multicolor menu from any other multicolor menu by hitting the appropriate key.

Copyright © 1984 by Microtechnic Solutions Inc.

SpryteByter

SHIFT Menu for Multicolor Sprites

A: up
B: down
C: right
D: left
E: invert
F: obvert
G: turn
H: draw
I: data
J: store

A: — shifts the sprite up 0 to 8 pixels.
B: — shifts the sprite down 0 to 8 pixels.
C: — shifts the sprite right 0 to 4 pixels.
D: — shifts the sprite left 0 to 4 pixels.
E: — turns the sprite upside-down.
F: — makes a mirror image of the sprite.
G: — same as 'turn' on the monocolor ROTATE menu except that the center of rotation is ALWAYS the center of the sprite. See notes on turn options on page 19.
H: — BRANCHES to SpryteByter's multicolor DRAW menu.
I: — BRANCHES to SpryteByter's multicolor DATA menu.
J: — stores the sprite in SpryteByter's memory.

NOTE: Functions A and B prompt the user with "How many rows?" Respond by entering a number from 0 to 8. Functions C and D prompt the user with "How many columns?" Respond by entering a number from 0 to 4.

Copyright © 1984 by Microtechnic Solutions Inc.

DATA Menu for Multicolor Sprites

A: I/O	E: movie
B: printout	F: draw
C: data in	G: shift
D: review	H: store

A: — allows you to save a file on tape or disk of all sprite data in SpryteByter's memory OR, if there are no sprites in SpryteByter's memory, allows you to load a sprite file into memory from tape or disk. See notes on page 23.

B: — prints a representation of the sprite and the data for the sprite if there is a printer (device#4) present. Does nothing if no printer is present. See notes on page 26.

C: — allows you to input sprite data and draws a sprite from that data. This allows you to preview and edit sprites from magazine, or other printed, program listings. Enter data as prompted with numbers from 0 to 255 and hit the RETURN key after each entry. If the function key f4 is hit instead of the RETURN key, then the typed in number will fill the rest of the sprite data. If you have not entered a number and hit the f8 key, then all remaining data numbers will become zero. See notes on page 25.

D: — accesses the REVIEW submenu.

E: — makes a movie of the sprites in SpryteByter's memory. Prompts for speed — enter a number from 0 (very slow) to 9.

F: — BRANCHES to SpryteByter's multicolor DRAW menu.

G: — BRANCHES to SpryteByter's multicolor SHIFT menu.

H: — stores the sprite in SpryteByter's memory.

Copyright © 1984 by Microtechnic Solutions Inc.

REVIEW Submenu for Multicolor DATA menu

sprite 0
A: next
B: back 1
C: edit
D: delete
E: del abov
F: insert
G: end reuv

A: — advances review to next sprite in memory.
B: — goes backward one sprite in memory.
C: — moves sprite under review into the sprite accumulator and returns to the DATA menu. Allows you alter sprites from memory.
D: — deletes sprite under review.
E: — deletes ALL sprites above sprite under review.
F: — inserts the accumulator sprite in memory in front of the sprite under review.
G: — returns to the DATA menu.

Copyright © 1984 by Microtechnic Solutions Inc.

SpryteByter Notes

General Information

The Sprite Accumulator: (Users not familiar with how data numbers determine sprites should read "Notes for Beginners" on page 28 before reading this section.) The SpryteByter program reserves 63 bytes of memory as a work area for sprite design which we refer to as the SPRITE ACCUMULATOR. Everything that is done to a sprite by the SpryteByter program is done in this accumulator. The information for the design grid and the four sprites seen in the upper righthand corner of most SpryteByter screens comes from this memory area. In the DATA menus, the area that usually displays the design grid is given over to a listing of the accumulator sprite data, the numbers that are actually stored in the sprite accumulator. In the REVIEW submenus, however, data which you have stored in SpryteByter's memory are displayed instead and the four sprites in the upper righthand corner of the screen are the sprites being reviewed. The accumulator sprite is displayed during REVIEW inside the rectangle with the data statements. This is the sprite which will be inserted into SpryteByter's memory if you use that option in the REVIEW submenus. This is also the sprite that is ANDed or ORed with the sprite under review if one of those options is used in the monocolor DATA/REVIEW submenu.

Options Available Only in the Monocolor Mode: Some, very useful, options available for designing monocolor sprites have not been found to be equally useful in designing multicolor sprites — tending to be more a cause for confusion than an aid. Therefore, the line draw options in the monocolor DRAW menu, the symmetry options in the monocolor ROTATE menu, and the logical AND and OR options in the monocolor DATA/REVIEW submenu have not been included for multicolor sprites. The "180 turn" option in the monocolor SHIFT menu can be accomplished in the multicolor SHIFT menu by choosing the "invert" and then "obvert" options.

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

The "turn' Options

Among the most useful features of SpryteByter are the turn options found in the monocolor ROTATE menu and the multicolor SHIFT menu. Follow these instructions for use of these options:

In the monocolor ROTATE menu, place the cursor at the point on the design grid around which you wish the sprite rotated. (The turn routine in the multicolor SHIFT menu ALWAYS rotates the sprite around the center of the sprite.)

When you press 'A' in the monocolor ROTATE menu or 'G' in the multicolor SHIFT menu, SpryteByter will prompt you with "Angle?" Respond with an integer in the range from -999 to 9999, then hit the RETURN key. Positive angles turn the sprite counter-clockwise; negative angles turn the sprite clockwise.

If you want a continuous turn for animation, enter an angle between 0 to 180 for a counter-clockwise turn or between 0 and -180 for a clockwise turn and then hit the f2 option key (i.e., the shifted f1 key) INSTEAD OF the RETURN key. If, for example, you enter 18 and then hit the f2 key, there will be a notice to wait and then SpryteByter will turn the sprite counter clockwise 18 degrees and store the result — then 36 degrees; 54 degrees; 72 degrees, 90 degrees, . . . , 180 degrees; storing all ten resulting sprites. If you enter -45 and then hit f2, SpryteByter will turn the sprite clockwise 45, 90, 135, and 180 degrees and store each of those four sprites. Hitting the reverse arrow key at the top lefthand corner of the keyboard will stop this turning after the current turn is finished and stored.

The rotation routines in SpryteByter have the unusual feature of taking into account the fact that the pixels (i.e., picture elements = dots) on your screen ARE NOT SQUARE, and thus the turns do not destroy the general shape integrity of your sprite. However, some loss of smoothness always results from ANY picture turning routine, and the question of what the most pleasing result would be is one of aesthetics, which cannot be resolved by computer programming. Therefore, turned sprites usually need some touching up by SpryteByter's user.

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

Sometimes the results of a rotation can be considerably improved by changing the center of rotation (for monochrome sprite) or by shifting the sprite up or down one row (for multicolor sprites). Note that the tiny X shaped cursor in the monocolor ROTATE menu marks the center of rotation and moves in increments of a half pixel. (The box shaped cursor in the DRAW and SHIFT menus move at whole pixel increments.) The results of a turn can in most cases be improved with dot by dot editing.

Do not overlook the convenience of the continuous turn feature implemented by pressing the f2 option key instead of the return key after the desired angle of rotation has been entered. This feature is a great patience preserver for designers of animated sprite programs. Note that in this routine, the original sprite that you are turning is NOT automatically stored in memory, so you may want to store it before RUNning the turn routine. Also note that the sprites automatically stored in this routine will likely need editing. View them using the movie option in the data menu to see if the turn is satisfactory for animation.

Copyright © 1984 by Microtechnic Solutions Inc.

Sprite Storage and Sprite Files

SpryteByter reserves 8K of memory for storage of sprites as you design them. In every SpryteByter menu there is an option to store the accumulator sprite. They are stored in numerical order from 0 to 127. At any time, the sprite file may be reviewed through the REVIEW submenus of the DATA menus.

If you wish to change a sprite in memory, it can be recalled to the accumulator through the edit option in the REVIEW submenus. If you wish to store a sprite in a particular location of memory, this can be done using the insert option in the REVIEW submenus. This is handy for arranging animations, which can be viewed in the DATA movie options, and for placing 31 sprites in early memory for use with The Game Maker.

There are two delete options in the REVIEW submenus. In the REVIEW submenus, if you press 'D' the sprite under review will be deleted. If you press 'E' all sprites in memory higher than the sprite under review will be deleted.

The I/O" Options

LOADing a Sprite File

To LOAD a sprite file into SpryteByter's memory follow these simple steps:

1. Branch to the DATA menu (in either the monocolor mode or the multicolor mode.)

2. Make sure that there are no sprites in SpryteByter's memory. If there are, they must all be deleted before SpryteByter will allow you to LOAD a file. To empty SpryteByter's memory, access the REVIEW option and choose sprite 0. Press 'E' and when SpryteByter prompts you with "Delete?" press 'Y' — this will delete from memory all sprites with numbers higher than 0. Now press 'D' and when SpryteByter

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

prompts you with "Delete it?" press 'Y' — this will delete sprite 0 from memory and SpryteByter will prompt you with "No sprite in memory! Press any key". Now, SpryteByter will return to the DATA menu and you are ready to LOAD a file.

3. Press 'A' to select the I/O option.
4. Your monitor screen will display
   1: LOAD from tape
   2: LOAD from disk
   R: return to DATA without LOAD

If you wish to LOAD a file from TAPE, press '1' and skip to instruction 6. If you wish to LOAD a file from disk, press '2'. (If you press 'R', SpryteByter will return to the DATA menu with no LOAD operation).

5. (DISK only) Your monitor screen will display
   D. Directory
   S. Disk status
   P. Proceed with LOAD
   R. Return without LOAD

If you press 'D', SpryteByter will print to the screen a catalog of the disk in your disk drive. Press any key to return to the LOAD submenu.

If you press 'S', SpryteByter will print the Disk Drive STATUS. Press any key to return to the LOAD submenu.

If you press 'R', SpryteByter will return to the DATA menu with out LOADing a file.

Press 'P' to proceed with a LOAD.

Skip to instruction 7.

6. (TAPE only) Your monitor screen will display

Ready the tape

Press 'r' to return to DATA
any other key to continue
LOAD

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

Follow these instructions.

7. Your monitor will display

Input file name:

File name 'r' will return to
DATA without LOAD

Type in the file name and hit RETURN. (The file name will be entered in lower case letters only. Pattern matching is allowed with disk. On tape, the first file on tape will be LOADed if no file name is specified.)

8. SpryteByter will now LOAD your file and return to the DATA menu.

SAVEing a Sprite File

To SAVE a sprite file into SpryteByter's memory follow these simple steps:

1. Branch to the DATA menu (in either the monocolor mode or the multicolor mode.)

2. Press 'A' to select the I/O option. NOTE: If there are no sprites in memory. SpryteByter will prepare for a LOAD instead of a SAVE.

3. Your monitor screen will display
   1: SAVE from tape
   2: SAVE from disk
   R: return to DATA without SAVE

If you wish to SAVE a file from TAPE, press '1' and skip to instruction 5. If you wish to SAVE a file from disk, press '2'. (If you press 'R', SpryteByter will return to the DATA menu with no SAVE operation).

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

4. (DISK only) Your monitor screen will display
   D. Directory
   S. Disk status
   P. Proceed with SAVE
   R. Return without SAVE

If you press 'D', SpryteByter will print to the screen a catalog of the disk in your disk drive. Press any key to return to the SAVE submenu.

If you press 'S', SpryteByter will print the Disk Drive STATUS. Press any key to return to the SAVE submenu.

If you press 'R', SpryteByter will return to the DATA menu without SAVEing a file.

Press 'P' to proceed with a SAVE.

Skip to instruction 7.

5. (TAPE only) Your monitor screen will display

Ready the tape

Press 'r' to return to DATA
any other key to continue
SAVE

Follow these instructions.

6. Your monitor will display

Input file name:

File name 'r' will return to
DATA without SAVE

Type in the file name and hit RETURN. (The file name will be entered in lower case letters only. Pattern matching is allowed with disk. On tape, the first file on tape will be SAVEd if no file name is specified.)

7. SpryteByter will now SAVE your file and return to the DATA menu.

Copyright © 1984 by Microtechnic Solutions Inc.

SpryteByter

The "data in" Options

The "data in" options are very easy to use. Experiment by entering these data

0, 0, 0
0, 0, 0
0, 255, 128
3, 255, 224
15, 231, 248
31, 239, 236
63, 255, 248
63, 255, 232
127, 255, 160
127, 252, 128
127, 240, 0
127, 252, 128
127, 255, 160
63, 255, 232
63, 255, 248
31, 255, 252
15, 255, 248
3, 255, 128
0, 255, 128

Now hit the f8 function key (SHIFT and f7) and your sprite is complete. Keep this sprite and read the section on "printout" below.

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

Printouts of Sprites

The printout routines from the DATA menus of SpryteByter print a representation of the accumulator sprite IF THERE IS A PRINTER, DEVICE #4, present, when, from a DATA menu you press 'B'. If device #4 is NOT present, pressing 'B' accomplished nothing.

The first time 'B' is pressed in a DATA menu in a given run of the SpryteByter program, you will be prompted with

Line feed?
(Y/N)

If you respond by pressing 'Y', SpryteByter will open a file numbered 128; if you respond 'N', SpryteByter will open a file numbered 4. For COMMODORE printers a response of 'N' is the correct response, for some other printers and interfaces a response of 128 is necessary to have a line feed for your printout. Consult your printer and interface manuals to decide which is the better response.

If you do not know the proper response for your printer, then answer 'Y' the first time you use this option on SpryteByter. If the printout is single spaced, then this was the proper response — continue to use it on subsequent uses of SpryteByter with the printer/interface. If the printout was double spaced then respond 'N' on subsequent uses of this option on SpryteByter.

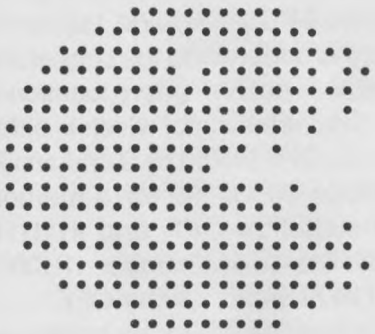
If you enter the data described in appendix Z and then use the printout option in the monochrome DATA menu your printout should look like this:

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

SpryteByter

MONOCHROME SPRYTEDATA



```
DATA  0 , 0 , 0
DATA  0 , 0 , 0
DATA  0 , 255 , 128
DATA  3 , 255 , 224
DATA  15 , 231 , 248
DATA  31 , 239 , 236
DATA  63 , 255 , 248
DATA  63 , 255 , 232
DATA  127 , 255 , 160
DATA  127 , 252 , 128
DATA  127 , 240 , 0
DATA  127 , 252 , 128
DATA  127 , 255 , 160
DATA  63 , 255 , 232
DATA  63 , 255 , 248
DATA  31 , 255 , 252
DATA  15 , 255 , 248
DATA  3 , 255 , 128
DATA  0 , 255 , 128
DATA  0 , 0 , 0
DATA  0 , 0 , 0
DATA  0 , 0 , 0
```

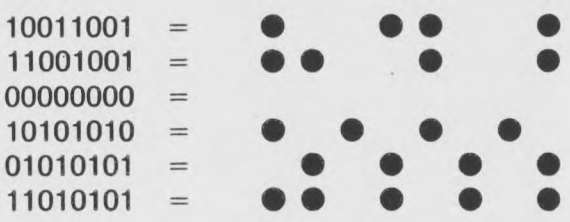
Copyright © 1984 by Microtechnic Solutions Inc.

Notes for Beginners

Sprites are defined by 63 numbers store in the Commodore 64's memory — each of these numbers being in the range from 0 to 255 decimal or 00000000 to 11111111 binary. In fact, EVERY memory location in the Commodore has in it a number in that range.

Although it is usually easier to think of a number in its decimal representation, for the purposes of understanding computer graphics, it is easier to think in terms of binary representations. For a discussion of how to convert decimal numbers to binary, see pages 76 and 77 of your Commodore 64 User's Guide, but for the moment, at least, it is not necessary to understand the conversion methods. It is sufficient now to think of each memory location in the computer as a sequence of 8 switches each of which is either turned ON or OFF. If we represent an OFF switch by 0 and an ON switch by 1, then the binary number 00000000 for a memory location represents all of the switches OFF and 11111111 represents all of the switches ON. The binary number 11110000 represents the first four switches ON and the last four OFF.

A monocolor sprite is a screen graphics character composed of a possible 504 dots on the screen, some turned on and some turned off. These are arranged in a rectangle 24 dots wide by 21 dots long. Each row (24 dots) is defined by three binary numbers in memory (3x8 = 24) with the 0's determining the OFF dots and the 1's determining the ON dots. (Usually these dots are referred to as pixels — short for picture elements.) So we can interpret the binary number 11111111 as a part of a sprite that looks like ********. Similarly,



etc.

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

Thus, an arrangement of 63 binary numbers like this

```
00000000,01111111,00000000
00000001,11111111,11000000
00000011,11111111,11100000
00000011,11100011,11100000
00000111,11011001,11110000
00000111,11011111,11110000
00000111,11011001,11110000
00000011,11100111,11100000
00000011,11111111,11100000
00000010,11111111,10100000
00000001,01111111,01000000
00000001,00111110,01000000
00000000,10011100,10000000
00000000,10011100,10000000
00000000,01001001,00000000
00000000,01001001,00000000
00000000,00111110,00000000
00000000,00111110,00000000
00000000,00111110,00000000
00000000,00011100,00000000
```

will result in a sprite which looks like this.

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

Now, look at SpryteByter (in the monocolor mode) and think of the plus signs in the grid as zeros and the reversed plus signs as ones and you will see how this data determines the sprite shapes.

The multicolor sprite mode is accessed from the monocolor COLOR menu. Multicolor sprites are a little different, but just about as easy to understand. Notice that the pixels in multicolor sprites are twice as big! This is because it takes two switches to determine each pixel. So a binary number such as 00100111, to determine four pixels. A 00 pair being OFF, a 10 pair causing the pixel to appear in the sprite color, a 01 pair causing the pixel to appear in multicolor 0, and a 11 pair causing the pixel to appear in multicolor 1. (If there are several multicolor sprites on the screen at one time, they may have different sprite colors, but all have the same multicolor 0 and multicolor 1).

It is not easy to see from the multicolor screen in SpryteByter how this works, so we have added a sort of third mode (besides monocolor and multicolor) to illustrate how these multicolor sprites are made up of binary numbers. This mode is a combination of the monocolor and multicolor mode for SpryteByter which gives you the monocolor menus and construction grid with multicolor sprites. To get to this peculiar mode first get SpryteByter into the multicolor data menu and SAVE a sprite. Then hit the 'E' key for "movie" and when you are prompted for speed DO NOT ENTER A NUMBER KEY but hit the SHIFT and RETURN keys together. This will put you in the MIXED mode in the data menu, from which you may branch to other monocolor menus. To escape this mixed mode, either go to the multicolor mode from the color menu or hit the SHIFT and RETURN keys together, which will put you in the monocolor mode.

Copyright © 1984 by Microtechnic Solutions Inc.

SpryteByter

Notes for Advanced Programmers

Our main considerations in writing SpryteByter have been to make a program useful to professional programmers and to make an entertaining program — in that order. Suggestions for improvements or useful enhancements are welcome.

Sprite files are saved by a machine code routine using the KERNAL SAVE at address $FFD8, and are completely relocatable. Sprites are saved in 64 byte blocks (63 bytes of sprite data and one byte unused) just as they are located in the COMMODORE 64 for program use. The Game Maker program first loads the sprites from a file into high memory and then relocates the first 31 sprites into sprite locations 33 through 63. (Sprite location N is the 63 bytes starting at 64*N.) This prevents the loading file from overwriting The Game Maker's basic. We have not used sprite location 32 because we feel that it is more desirable to load programs in the usual place for basic programs (beginning at $0800) and have them run with a RUN command instead of a SYS. We stopped at location 63 because sprite locations 64 through 127 are not readily available or easily usable — especially for Basic programmers.

The sprite files are program files. As has been indicated, the sprite files can be located anywhere (with obvious exceptions such as page 1) using the KERNAL LOAD at $FFD5. A possible and useful location is starting with sprite location 128, i.e., at $2000. We have been (nearly) successful loading a sprite file here simply by typing

POKE43,0:POKE44,32:LOAD"⟨filename⟩",8

which gives us a successful load EXCEPT that the first three bytes of the first sprite are changed. This is easy enough to repair. We consider it expedient after such a load to remember immediately to

POKE43,1:POKE44,8.

Copyright © 1984 by Microtechnic Solutions Inc.

The Game Maker

The Game Maker is a SpryteByter bonus program which provides an excellent format for programming games on the Commodore 64 using up to 31 different sprite definitions — without the necessity of data statements for those sprites. To use The Game Maker, follow these steps:

1. Design sprites for your program using SpryteByter and SAVE them through the I/O option in the data menu.
2. LOAD The Game Maker.
3. Prepare the disk or tape with your sprite file for LOAD.
4. Run Game Maker. The Game Maker will ask whether you are LOADING with tape or disk. Respond accordingly.
5. When prompted to do so, input the file name and hit the RETURN key. (As usual, the file name is not necessary on tape LOADs if you wish to LOAD the first file on the tape. Simply hit the RETURN key when you are certain that the tape is ready to LOAD.) The Game Maker will LOAD your sprite file and create a new program format — ready for you to program your game.

You should now see, on the monitor screen, the number 33 and your first sprite. Hit 'A' and you will see the number 34 and your second sprite. Hit 'A' again and advance to 35 and your third sprite — and so on through all the sprites from your file (to a maximum of 31.) If your file had more than 31 sprites, only the first 31 of them will go into The Game Maker program — the last will be number 63). If you hit 'B' the review will back up as in the REVIEW menus of SpryteByter. The numbers by your sprites in this review are the sprite definition locations you will use in your game program for reading the sprites. For example, if, in your program, you POKE

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

2040,33, then sprite #0 will be defined by the information at location 33 and will be the first sprite from your file. Please see the COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE for details on programming with sprites.

6. Hit the RUN/STOP and RESTORE keys and then list. You are now ready to program your game, using line numbers less than 60000. You may want to make line 59999 either a STOP or END statement.

7. To SAVE your game, type 'RUN61000' and hit the RETURN key. DO NOT SAVE with a SAVE command as this will result in your SAVEing the program WITHOUT the sprite information.

8. To play your game, just LOAD and RUN.

9. If, while programming, you wish to review the sprites in memory, type RUN60000 and hit the RETURN key.

If, in the course of programming your game, you want to revise your sprites, a simple way is to follow these steps:

1. Revise your sprite file using SpryteByter.

2. LOAD your game program and run it. Then SAVE your game in a temporary file USING THE SAVE COMMAND.

3. Type POKE44,8:POKE2048,O:CLR and hit the return key.

4. LOAD "GAME MAKER" and RUN as if you plan to make a new game. LOAD your revised sprite file through The Game Maker.

5. Hit the RUN/STOP and RESTORE keys.

6. LOAD your temporary file.

7. SAVE by RUNning 61000. Your newly SAVEd program will now contain the new sprites.

►►►

Copyright © 1984 by Microtechnic Solutions Inc.

If, by accident, you SAVE your game with the SAVE command and lose your sprites, you can recover them. Simply use your incorrectly SAVEd program as the temporary file together with your original sprite file and proceed as if you were revising the sprites in your game (as described above.)

The ADVANCED basic programmer can modify existing games to eliminate data statements by merging the program into The Game Maker format. No one general method, however, applies to all programs. ASSEMBLY LANGUAGE programmers may want to convert sprite files from program files to sequential files — refer to your disk drive manual.

We look forward to seeing many exciting games developed with the aid of SpryteByter and The Game Maker.

BACKUPS AND WARRANTY

A backup copy of SpryteByter may be obtained for $10.00 (check or money order — do not send cash via mail). Warranty card on file or proof of purchase required.

MICROTECHNIC SOLUTIONS, INC.
P.O. Box 2940
New Haven, CT 06515

Microtechnic Solutions warrants the SpryteByter software diskette for 90 days, provided it has not been physically abused or subjected to magnetic fields. Replacements after the warranty period are charged at $5.00.

Copyright © 1984 by Microtechnic Solutions Inc.

NOTES:

COPYRIGHT NOTICE

COPYRIGHT © 1984 by Microtechnic Solutions, Inc. All rights reserved. No part of this publication may be reproduced in whole or in part without the prior written consent of Microtechnic Solutions, Inc.

MICROTECHNIC SOLUTIONS INC.
P.O. BOX 2940, New Haven, Ct. 06515
Telephone: (203) 389-8383

MTS-UG. 4/84 *PRINTED IN U.S.A.*